

## EXECUTION PROCESSOR CHANGE SYSTEM FOR JOB

**Publication number:** JP8030471 (A)

**Also published as:**

**Publication date:** 1996-02-02

US5870604 (A)

**Inventor(s):** YAMAGISHI TADASHI

**Applicant(s):** HITACHI LTD

**Classification:**

- international: G06F15/16; G06F9/46; G06F9/50; G06F15/177; G06F15/16;  
G06F9/46; (IPC1-7): G06F9/46; G06F15/16

- European: G06F9/46A4L

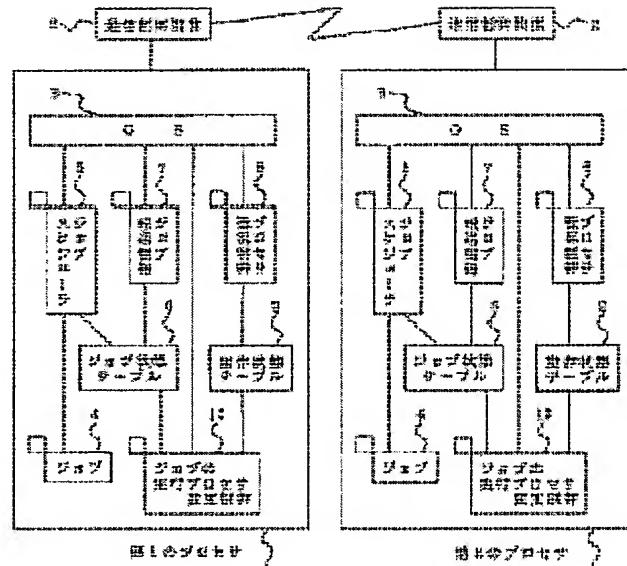
**Application number:** JP19940161827 19940714

**Priority number(s):** JP19940161827 19940714

### Abstract of JP 8030471 (A)

**PURPOSE:** To decentralize a load to processors in a system consisting of the processors which execute jobs by changing an execution processor for a job waiting to be executed into a light-load processor.

**CONSTITUTION:** When a job 4 is inputted to a processor 1, a job scheduler 5 stores information on the job in a job state table 6 and executes the job in order. A processor monitor and control part 8 reports the load state of the processor 1 by using a communication controller 2 and stores it in an operation state table 9. A job execution processor change control part 10 changes a job in the job state table 6, which waits to be executed, to a light-load processor in the operation state table 9. A job transfer control part 7 transfers the job to the execution processor for the job.



Data supplied from the **esp@cenet** database — Worldwide

(19)日本国特許庁 (JP)

# (12) 公開特許公報 (A)

(11)特許出願公開番号

特開平8-30471

(43)公開日 平成8年(1996)2月2日

(51)Int.Cl.<sup>6</sup>  
G 0 6 F 9/46  
15/16

識別記号 3 4 0 D 7737-5B  
3 8 0 Z

F I

技術表示箇所

審査請求 未請求 請求項の数8 O.L. (全9頁)

(21)出願番号

特願平6-161827

(22)出願日

平成6年(1994)7月14日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 山岸 正

神奈川県横浜市戸塚区戸塚町5030番地 株

式会社日立製作所ソフトウェア開発本部内

(74)代理人 弁理士 小川 勝男

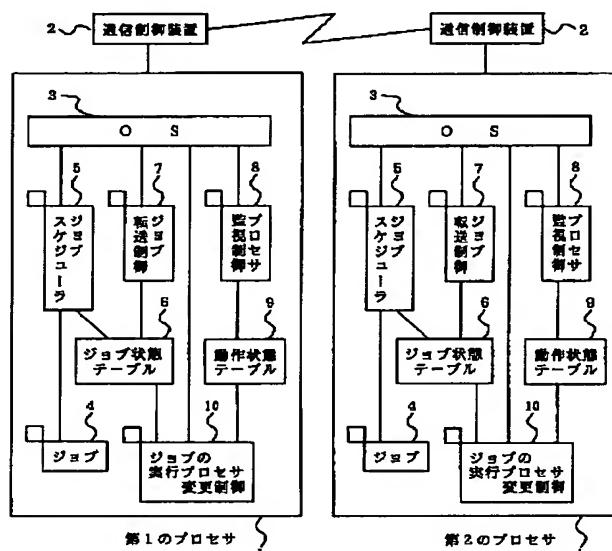
(54)【発明の名称】 ジョブの実行プロセサ変更方式

(57)【要約】

【目的】ジョブを実行する複数のプロセサで構成されるシステムにおいて、実行待ちのジョブの実行プロセサを負荷の低いプロセサに変更することによりプロセサ間の負荷分散を実現する。

【構成】プロセサ1にジョブ4が入力されると、ジョブスケジューラ5がジョブ状態テーブル6にジョブの情報を格納してジョブを順番に実行する。プロセサ監視制御部8がプロセサ1の負荷状態を通信制御装置2を使い相互に通知して、動作状態テーブル9に格納する。ジョブの実行プロセサ変更制御部10がジョブ状態テーブル6の中の実行待ちのジョブを動作状態テーブル9の中の負荷の低いプロセサに変更する。ジョブ転送制御部7はジョブの実行プロセサへジョブを転送する。

図1



## 【特許請求の範囲】

【請求項1】ジョブを実行する複数のプロセサと、前記ジョブの実行順序を制御するジョブスケジューラと、前記複数のプロセサ間のデータ転送を行う通信制御装置と、前記複数のプロセサ間でジョブの転送を行うジョブ転送手段とで構成されるシステムにおいて、前記複数のプロセサ間でプロセサの負荷状態を相互に監視するプロセサ監視手段と、前記複数のプロセサ間でジョブの実行プロセサを相互に変更するジョブの実行プロセサ変更手段とを具備したことを特徴とするジョブの実行プロセサ変更方式。

【請求項2】請求項1において、前記ジョブの実行プロセサ変更手段は、第1のプロセサのジョブスケジューラが実行させるジョブが無くなった時に、より負荷の高い第2のプロセサに対してジョブの実行プロセサの変更要求を送信する手段であることを特徴とするジョブの実行プロセサ変更方式。

【請求項3】請求項1において、前記ジョブの実行プロセサ変更手段は、第1のプロセサからジョブの実行プロセサの変更要求を受信した時に、第2のプロセサでジョブの実行プロセサを変更する手段であることを特徴とするジョブの実行プロセサ変更方式。

【請求項4】請求項1において、ジョブの実行プロセサ変更手段は、第1のプロセサの負荷が高い時に、第1のプロセサの実行待ちジョブの実行プロセサを、負荷の低い第2のプロセサに変更する手段であることを特徴とするジョブの実行プロセサ変更方式。

【請求項5】請求項1において、負荷はプロセサの実行待ちジョブ数であることを特徴とするジョブの実行プロセサ変更方式。

【請求項6】請求項1において、負荷はプロセサのCPU使用率であることを特徴とするジョブの実行プロセサ変更方式。

【請求項7】請求項1において、負荷はプロセサのI/O使用率であることを特徴とするジョブの実行プロセサ変更方式。

【請求項8】請求項1において、負荷はプロセサのメモリ使用率であることを特徴とするジョブの実行プロセサ変更方式。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】本発明は、計算機システムでジョブを実行させる場合の実行プロセサの決定方式について、特に計算機システムが複数台のプロセサから構成される場合に好適なジョブの実行プロセサ変更方式に関する。

## 【0002】

【従来の技術】従来の方式では、例えば特開平4-335442号公報に記載のように、実行待ちのジョブのクラスを最も負荷の低いクラスに変更することにより、シ

ステムの負荷分散を実現していた。

## 【0003】

【発明が解決しようとする課題】上記従来技術は、ホスト計算機が複数のプロセサから構成される場合について考慮されておらず、ジョブクラスを変更しても実行プロセサが変更できないため、特定のプロセサに処理が集中するという課題があった。

【0004】また、ジョブが入力された時にプロセサの負荷を考慮して実行プロセサを決定する方式では、ジョブが実行待ちの間にプロセサ間の負荷バランスがくずれてしまうという問題があった。

【0005】本発明の目的は、プロセサ間でプロセサの負荷状態を相互に監視する手段を設け、ジョブが実行待ちの時にプロセサの負荷を考慮して実行プロセサを変更する手段を設けることにより、オペレータ等の入手を介したり、特別なジョブスケジューラを設けたりする事なく、プロセサ間の負荷分散を実現できるジョブの実行プロセサ変更方式を提供することにある。

## 【0006】

【課題を解決するための手段】上記目的は、ジョブを実行する複数のプロセサと、ジョブの実行順序を制御するジョブスケジューラと、プロセサ間のデータ転送を行う通信制御装置と、プロセサ間でジョブの転送を行うジョブ転送手段で構成されるシステムにおいて、プロセサ間でプロセサの実行待ちジョブ数、CPU使用率、I/O使用率及びメモリ使用率の負荷状態を相互に監視するプロセサ監視手段と、プロセサ間で第1のプロセサのジョブスケジューラが実行させるジョブが無くなった時に、負荷の高い第2のプロセサに対してジョブの実行プロセサの変更要求を送信するジョブの実行プロセサ変更手段と、第1のプロセサからジョブの実行プロセサの変更要求を受信した時に、第2のプロセサでジョブの実行プロセサを変更するジョブの実行プロセサ変更手段とを設けることにより達成される。

【0007】さらに、プロセサ間で第1のプロセサの負荷が高い時に、第1のプロセサの実行待ちジョブの実行プロセサを負荷の低い第2のプロセサに変更するジョブの実行プロセサ変更手段を設けることにより達成される。

## 【0008】

【作用】プロセサ監視手段は、プロセサの実行待ちジョブ数、CPU使用率、I/O使用率、メモリ使用率等の負荷状態を単位時間毎に調べ、動作状態が変化した場合、動作状態を他のプロセサへ通知する。

【0009】第1のプロセサのジョブの実行プロセサ変更手段は、第1のプロセサのジョブスケジューラが実行させるジョブが無くなった時に、プロセサ監視手段が取得した他のプロセサの動作状態を調べ、負荷の高い第2のプロセサに対してジョブの実行プロセサの変更要求を送信する。ジョブの実行プロセサの変更要求を受信した

第2のプロセサのジョブの実行プロセサ変更手段がジョブの実行プロセサを変更することにより、ジョブ転送手段がジョブを第1のプロセサへ自動的に転送する。

【0010】また、負荷が高い第1のプロセサのジョブの実行プロセサ変更手段は、第1のプロセサの実行待ちジョブの実行プロセサを負荷の低い第2のプロセサに変更することにより、ジョブ転送手段がジョブを第2のプロセサへ自動的に転送する。

【0011】このため、ユーザはジョブを実行するのに適切なプロセサを選択できる。また、プロセサの負荷分散が図れるため、センタ管理者は、プロセサの有効利用が図れる。

【0012】

【実施例】以下に、本発明の一実施例を図面によって詳細に説明する。

【0013】図1は本発明の一実施例のジョブの実行プロセサ変更方式の構成例である。

【0014】図1において、1はジョブを実行する複数のプロセサ、2はプロセサ間のデータ転送を行う通信制御装置、プロセサ1と通信制御装置2は入出力チャネルにより接続されている。プロセサ間を接続する通信制御装置2は互いに通信回線により接続されている。

【0015】図1のプロセサ1のもとでは、プロセサ1の動作を制御するオペレーティングシステム(OS)3、OS3のもとでデータ処理を行うジョブ4、ジョブ4の実行順序を制御するジョブスケジューラ5、ジョブスケジューラ5がプロセサに指定されたジョブ実行多重度の数だけジョブを順番に実行させていくためにジョブの実行制御情報を格納したジョブ状態テーブル6、ジョブ状態テーブル6のジョブの実行プロセサ情報が他のプロセサの場合にジョブを他のプロセサへ転送するジョブ転送制御部7、プロセサの動作状態を調べプロセサ間で相互に動作状態を転送するプロセサ監視装置8、プロセサ監視装置8が取得したプロセサの動作状態を格納した動作状態テーブル9、プロセサの動作状態からジョブの実行プロセサを変更するジョブの実行プロセサ変更制御部10が存在する。

【0016】図2から図3は本発明によるジョブの実行プロセサ変更方式のデータ構成例である。

【0017】図2は図1のジョブ状態テーブル6のデータ構成例である。

【0018】ジョブ状態テーブル6は、ジョブを識別するためのジョブ名6a、ジョブの入力時に決定したジョブの実行プロセサ6b、ジョブが実行中か実行待ちかを表すジョブの動作状態6c、ジョブ制御文で指定されたジョブの実行可能プロセサ6dからなっている。

【0019】図3は図1のプロセサ1に作成した動作状態テーブル9のデータ構成例である。動作状態テーブル9は、プロセサを識別するためのプロセサ名9a、プロセサ毎の負荷を示す実行待ちジョブ数9b、CPU使

用率9c、I/O使用率9d、メモリ使用率9eからなっている。実行待ちジョブ数9bは、値が「1」以上の場合は実行待ちのジョブ数を表している。値が「0」の場合は、実行待ちのジョブがなくプロセサに指定されたジョブ多重度と同数のジョブが実行中であることを表している。値が「-1」以下の場合は、プロセサに指定されたジョブ多重度に対して実行が不足しているジョブ数を表している。CPU使用率9c、I/O使用率9d、メモリ使用率9eは、単位時間ごとの各使用率を百分率で表したものである。

【0020】図4から図7は本発明によるジョブの実行プロセサ変更方式の動作手順例である。

【0021】まず、図4は図1のプロセサ1に作成したジョブスケジューラ5の動作手順例である。

【0022】ジョブスケジューラ5はプロセサ1の開始時にOS3から起動され、ジョブ状態テーブル6の作成を行い(5a)、OS3からの要求を待つ(5b)。OS3からの要求がジョブの入力要求の場合(5c)は、ジョブ制御文を解釈してジョブを実行可能なプロセサの中から負荷(実行待ちジョブ数、CPU使用率、I/O使用率、メモリ使用率)が少ないプロセサをジョブの実行プロセサとして選び、ジョブ実行制御情報をジョブ状態テーブル6の先頭から順番に格納していく(5d)。OS3からの要求がジョブ終了通知の場合(5e)は、ジョブ状態テーブル6から終了したジョブのジョブ実行制御情報を削除する(5f)。以上の処理によりプロセサに指定されたジョブ実行多重度に対して実行中のジョブ数が不足している場合(5g)は、不足している数だけジョブ状態テーブル6の先頭から順番に実行待ちジョブを実行させる(5h)。プロセサ1の動作終了通知を受けた場合(5i)は、動作を終了する。プロセサ1が動作を続行する場合は(5b)の処理にもどる。

【0023】次に、図5は図1のプロセサ1に作成したジョブ転送制御部7の動作手順例である。

【0024】ジョブ転送制御部7はプロセサ1の開始時にOS3から起動され、OS3からの要求を待つ(7a)。OS3からの要求が他のプロセサへのジョブ転送要求の場合(7b)は、ジョブ状態テーブル6を参照してジョブの動作状態6cが実行待ちで実行プロセサ6bが他プロセサとなっているジョブを転送する。プロセサ1の動作終了通知を受けた場合(7d)は、動作を終了する。プロセサ1が動作を続行する場合は(7a)の処理にもどる。

【0025】図6は、図1のプロセサ1に作成したプロセサ監視制御部8の動作手順例である。

【0026】プロセサ監視制御部8はプロセサ1の開始時にOS3から起動され、動作状態テーブル9の作成を行い(8a)、OS3からの要求を待つ(8b)。OS3からの要求が単位時間経過通知の場合(8c)は、プロセサの実行待ちジョブ数、CPU使用率、I/O使用

率、メモリ使用率をOS3から取得し動作状態テーブル9の自プロセサの動作状態を変更する(8d)。以上の処理により動作状態テーブル9を変更した場合(8e)は、他プロセサへ自プロセサの動作状態の変更を通知する(8f)。OS3からの要求が他プロセサから動作状態変更通知の場合(8g)は、動作状態テーブル9の他プロセサの状態を変更する(8h)。プロセサ1の動作終了通知を受けた場合(8i)は、動作を終了する。プロセサ1が動作を続行する場合は(8b)の処理にもどる。

【0027】図7は、図1のプロセサ1に作成したジョブの実行プロセサ変更制御部10の動作手順例である。

【0028】実行プロセサ変更制御部10はプロセサ1の開始時にOS3から起動され、OS3からの要求を待つ(10a)。OS3からの要求がジョブ終了通知の場合(10b)は、動作状態テーブル9を参照して自プロセサの実行待ちジョブ数が0の場合は、他プロセサの中で実行待ちジョブ数が多い他プロセサの実行プロセサ変更制御部に対して、他プロセサの実行待ちジョブを自プロセサへ実行プロセサを変更することを要求する。変更不可能応答があった場合は、ジョブの実行プロセサ変更要求を実行待ちジョブ数が多いプロセサへ順番に行う(10c)。

単位時間経過通知の場合(10d)は、ジョブ状態テーブル6を先頭から参照してジョブの動作状態6cが実行待ちになっているジョブの中で実行可能プロセサ6dが他のプロセサで実行可能なジョブを選び、動作状態テーブル9を参照して実行可能な他のプロセサ9aの実行待ちジョブ数9bが「-1」以下になっているプロセサ(プロセサに指定されたジョブ多重度に対してジョブの実行が不足しているプロセサ)名に実行プロセサ6bの領域を変更して、ジョブ転送制御部へ連絡する(10e)。なお、変更可能なプロセサ名が複数ある場合は、負荷(CPU使用率、I/O使用率、メモリ使用率)が少ないプロセサを選び、ジョブの実行プロセサ名を変更する。OS3からの要求が他プロセサからのジョブの実行プロセサ変更要求の場合(10f)は、ジョブ状態テーブル6を先頭から参照してジョブの動作状態6cが実行待ちになっているジョブの中で実行可能プロセサ6dに変更要求元のプロセサ名が含まれているジョブを一つ選び、ジョブの実行プロセサ9bを変更要求元のプロセサ名に変更してジョブ転送制御部へ連絡と共に変更要求プロセサへ変更可能応答を返す。実行プロセサを変更不可能な場合は、要求プロセサへ変更不可能

応答を返す(10g)。プロセサ1の動作終了通知を受けた場合(10h)は、動作を終了する。プロセサ1が動作を続行する場合は(10a)の処理にもどる。

【0029】以上の処理によって、ジョブ依頼者がジョブの実行プロセサを意識することなく、各プロセサの動作状態が適正になるようにジョブを実行させるプロセサを自動的に変更することができる。

【0030】

【発明の効果】以上説明したように、本発明によれば、複数のプロセサの中から、プロセサの動作状態に従い、実行待ちのジョブの実行プロセサを自動的に変更するため、センタ管理者はプロセサの有効利用が図れるという効果がある。

【0031】また、各プロセサの動作状態が適正になるようにジョブの実行プロセサを自動的に変更するため、ユーザはジョブを実行するのに適切なプロセサが選択できるという効果がある。

【図面の簡単な説明】

【図1】本発明によるジョブの実行プロセサ変更方式の構成例を示す図である。

【図2】図1のジョブ状態テーブル6のデータ構成例を示す図である。

【図3】図1の動作状態テーブル9のデータ構成例を示す図である。

【図4】図1のジョブスケジューラ5の動作手順例を示すフローチャートである。

【図5】図1のジョブ転送制御部7の動作手順例を示すフローチャートである。

【図6】図1のプロセサ監視制御部8の動作手順例を示すフローチャートである。

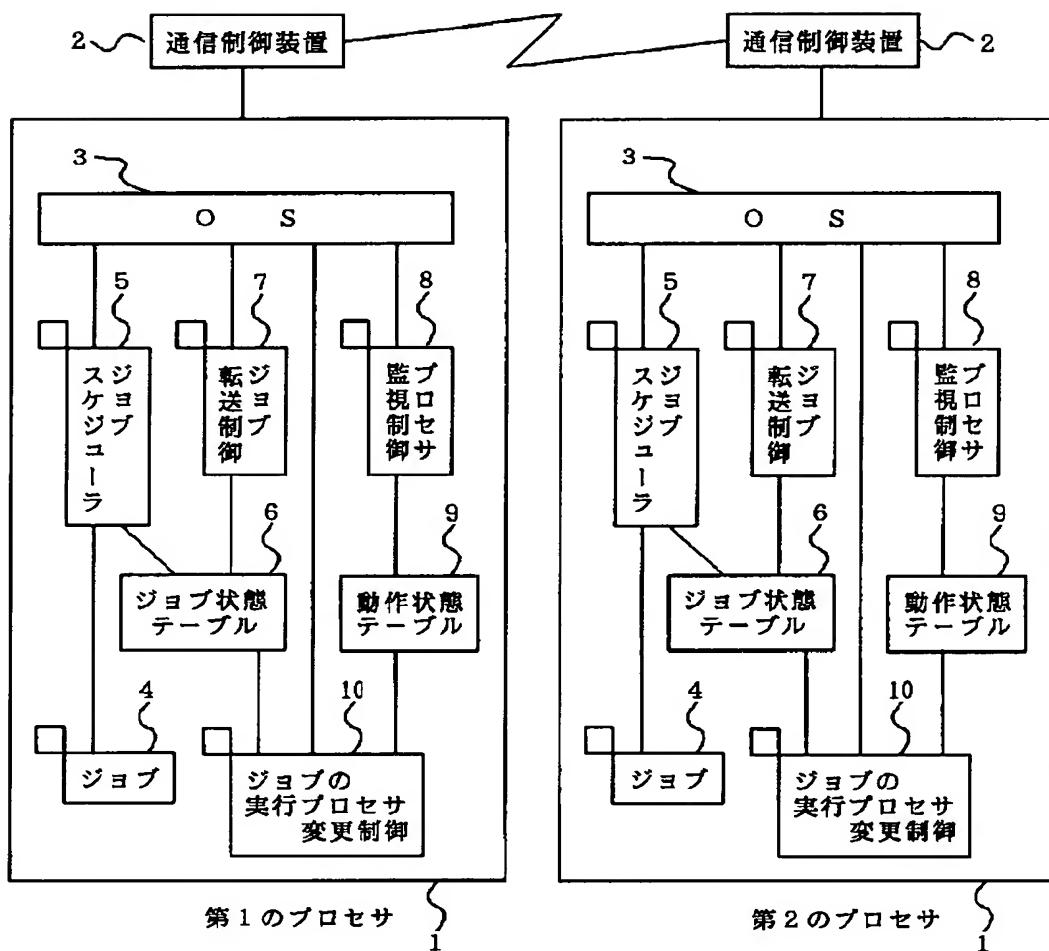
【図7】図1のジョブの実行プロセサ変更制御部10の動作手順例を示すフローチャートである。

【符号の説明】

- 1・・・プロセサ,
- 2・・・通信制御装置,
- 3・・・オペレーティングシステム(OS),
- 4・・・ジョブ,
- 5・・・ジョブスケジューラ,
- 6・・・ジョブ状態テーブル,
- 7・・・ジョブ転送制御部,
- 8・・・プロセサ監視制御部,
- 9・・・動作状態テーブル,
- 10・・・ジョブの実行プロセサ変更制御部。

【図1】

図1



【図2】

図2

図2は「ジョブの実行プロセサ変更制御」(10)に表示される「ジョブの実行プロセサ変更制御」(10)用のデータ表です。

表の構成：

- 列名：ジョブ名 (6a), 実行プロセサ (6b), ジョブの動作状態 (6c), 実行可能プロセサ (6d)
- データ：

ABCDEF	P1	実行中	P1, P2, P3
XYZ000	P1	実行中	P1
ABC000	P1	実行待ち	P1, P2
A12345	P1	実行待ち	P1, P2, P3
XYZ123	P1	実行待ち	P1, P2

【図3】

図3

図3は「CPU使用率」(8c)用のデータ表です。

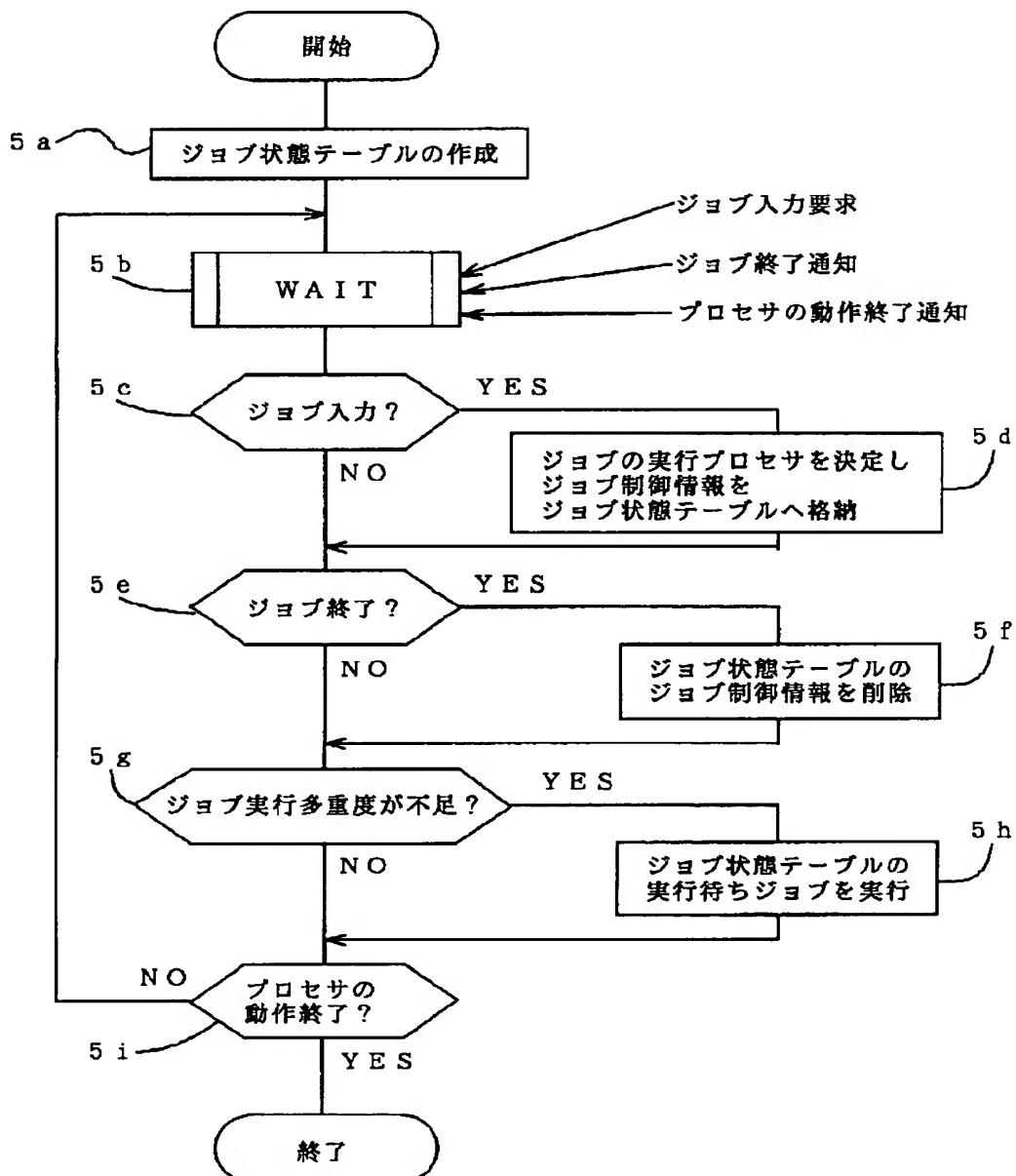
表の構成：

- 列名：プロセサ名 (8a), 実行待ちジョブ数 (8b), CPU使用率 (8c), I/O使用率 (8d), メモリ使用率 (8e)
- データ：

P1	1.5	80	30	50
P2	-1	50	20	80
P3	4	40	60	20
P4	-2	60	20	40

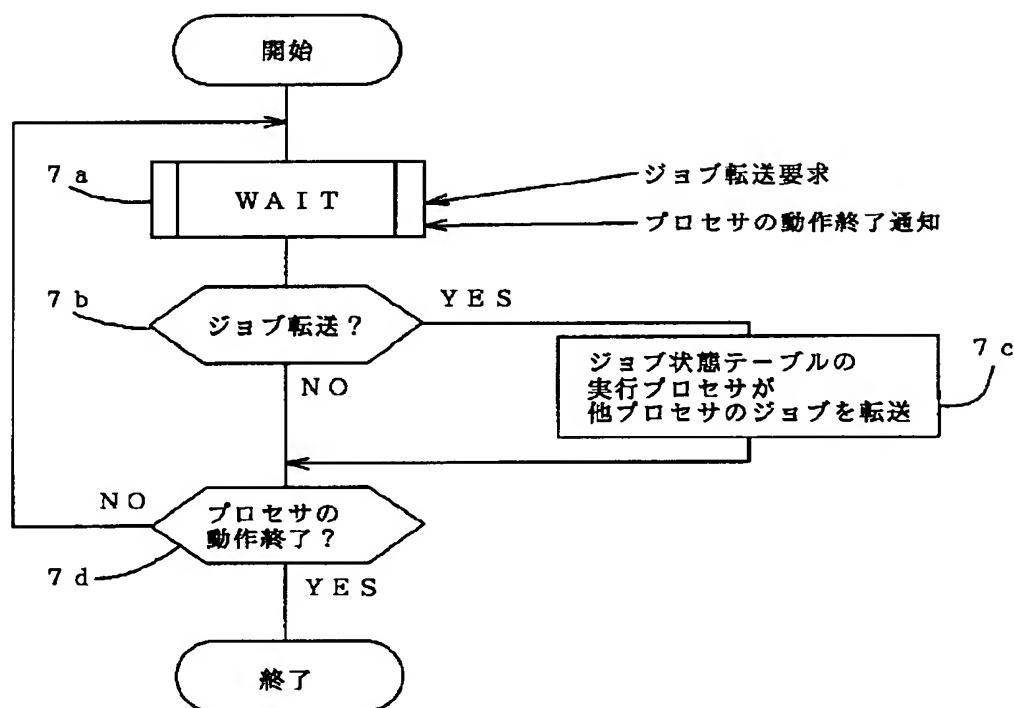
【図4】

図4



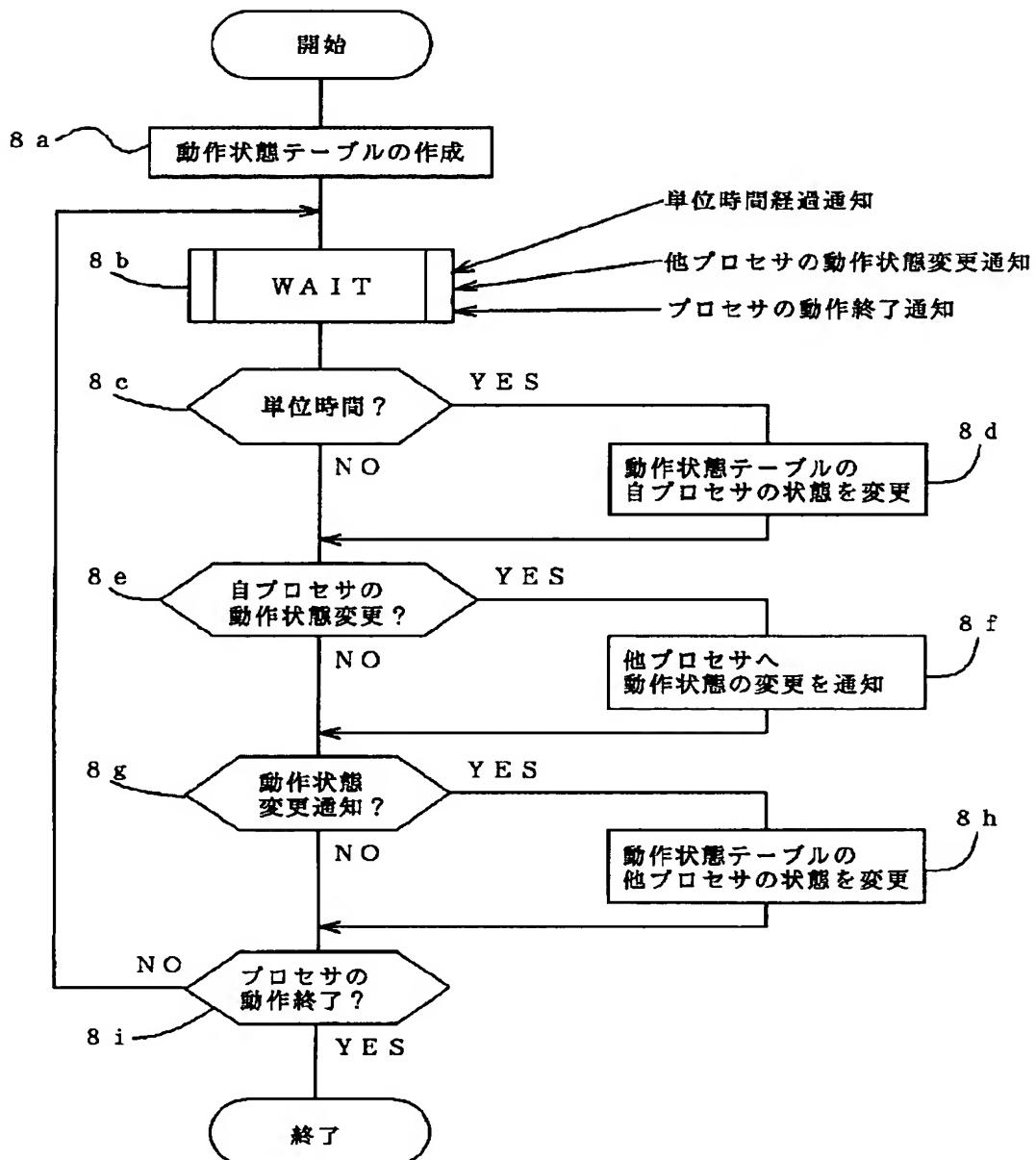
【図5】

図5



【図6】

図6



【図7】

図7

